Learning to Target: What Works for Behavioral Targeting

Sandeep Pandey[◇§], Mohamed Aly[◇§], Abraham Bagherjeiran[*◇†], Andrew Hatch[◇§],
Peter Ciccolo[*‡], Adwait Ratnaparkhi[*♯], Martin Zinkevich[§]

§ Yahoo! Research, 4301 Great America Parkway, Santa Clara, CA 95054, USA
† ThinkersR.Us Inc., 1262 Socorro Avenue, Sunnyvale, CA 94089, USA
‡ Google Inc., 1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA
♯ 33Across Inc., 440 N. Wolfe Road, Sunnyvale, CA 94085, USA

{spandey | aly}@yahoo-inc.com | abagher@thinkersr.us | aohatch@yahoo-inc.com |
ciccolo@google.com | adwait.ratnaparkhi@33across.com | maz@yahoo-inc.com

ABSTRACT

Understanding what interests and delights users is critical to effective behavioral targeting, especially in information-poor contexts. As users interact with content and advertising, their passive behavior can reveal their interests towards advertising. Two issues are critical for building effective targeting methods: what metric to optimize for and how to optimize. More specifically, we first attempt to understand what the learning objective should be for behavioral targeting so as to maximize advertiser's performance. While most popular advertising methods optimize for user clicks, as we will show, maximizing clicks does not necessarily imply maximizing purchase activities or transactions, called *conversions*, which directly translate to advertiser's revenue. In this work we focus on conversions which makes a more relevant metric but also the more challenging one. Second is the issue of how to represent and combine the plethora of user activities such as search queries, page views, ad clicks to perform the targeting. We investigate several sources of user activities as well as methods for inferring conversion likelihood given the activities. We also explore the role played by the temporal aspect of user activities for targeting, e.g., how recent activities compare to the old ones. Based on a rigorous offline empirical evaluation over 200 individual advertising campaigns, we arrive at what we believe are best practices for behavioral targeting. We deploy our approach over live user traffic to demonstrate its superiority over existing state-of-the-art targeting methods.

Categories and Subject Descriptors

H.4.m [**Information Systems**]: Miscellaneous

General Terms

Algorithms, Performance, Experimentation

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

Keywords

Behavioral targeting, user modeling, advertising

1. INTRODUCTION

Advertisers want to spend the smallest amount of money to get the largest increase in profit, irrespective of the form of advertising they are involved in such as newspapers, magazines, television (*offline advertising*), and internet *(online advertising*). However, online advertising provides advertisers with more immediate feedback (when users click on their ads and visit their web pages) and publishers have greater knowledge of their users (demographic information and past behavior). For example, suppose that you want to sell trucks. You could advertise during baseball games and football games on television, hoping that people watching the game want a truck. If you advertise online, you could put your truck advertisements next to sports stories on an online publisher's site. Also, you could have your advertisements appear in "Car and Driver" or on "Yahoo Autos" (where the content is related to the ad). But the more remarkable power of online advertising is that you could show an advertisement next to a sports page only for people who have recently visited Yahoo Autos. This is *behavioral targeting.*

The key behind behavioral targeting is that the advertisers can show ads only to users within a specific demographic of high value (such as people likely to buy a car) and combine that with a larger number of opportunities (places to show ads) per user. Moreover, as we gather more information about a user, we can provide them with a better experience in every future interaction.

In general, targeting methods match the users within a given context to an appropriate ad. The context of the user consists of the page the user is currently visiting, the time, and the user's historical online behavior. Three types of targeting methods are popular in the advertising industry: *property*, *user segment*, and *behavioral targeting.* Property targeting refers to placing ads on specific web pages where interested users will appear, such as showing online brokerage ads on financial related pages. Although this reaches users who visit these finance pages, it may miss users who

◇ The first four authors have contributed equally to the paper.
* Work done while at Yahoo!.

use some other web sites for their financial information. User segment targeting focuses on the gender and age of a user, and is only capable of targeting broad groups. Behavioral targeting involves using historical online information about the user to aid the publisher in showing them relevant ads wherever they appear. Whereas property targeting targets pages, and user segment targeting targets generic groups, behavioral targeting targets individuals.

Clearly, understanding user interests is critical to effective behavioral targeting. As users interact with content and advertising, their passive behavior can reveal their interests towards advertising. Exploiting this behavior at a large scale is the goal of behavioral targeting solutions. In this paper we frame the targeting problem as an optimization problem and describe a machine learning based approach for solving it. We tackle with two key issues in doing so. First, it is important to optimize behavioral targeting so that advertisers get most bang for their buck. Most existing work in behavioral targeting uses clicks on ads as a proxy of interest [4, 11, 14]. Clicks are used simply because they are available and other information is not available at a large scale. Recently, however, advertisers have been willing to share feedback at the level of individual users, telling publishers which of the users who saw the ad have actually purchased the product [1, 2, 3, 6, 12]. Since conversions are the ultimate goal of advertisers, we test whether models based on conversions or clicks better predict whether a conversion eventually occurs. In particular, we show that maximizing for clicks does not lead to maximizing for conversions, hence providing evidence for the necessity of developing models that are specifically optimized for conversions.

Conversions are so rare that developing models for them presents further challenges, which is the second issue we address in this paper. Rarity of conversions forces us to parsimoniously mine the user historical online behavior. Several questions are worth investigating here. For example, how should we represent and combine different user activities such as search queries, page views and ad clicks? Which activities are more indicative of user's interests, browsed pages or issued queries? Does more activity on a certain topic imply more chances of converting? How does the temporal aspect of user behavior (i.e., timestamps associated with the activities) relate to conversion likelihood?

Our main contributions in this paper are as follows:

- We provide insights into which types of user behavior and activities (e.g., page views, search logs, ad views/clicks, *etc.*) are informative. See Section 4.3.
- We give methods for how to translate user activities into features which work best for capturing ad-relevant information in user behavior. See Section 4.4.
- We show that although conversions are more rare, they can be more informative for targeting than clicks when available. Also, we describe how clicks can be used in conjunction with conversions to improve the targeting. See Section 5.
- We look into understanding the temporal aspect of user history and study how recency of a user activity and length of user history affect the targeting performance. See Section 6.
- We perform extensive experiments using data from more than 200 display advertising campaigns to arrive at what we believe are best practices for behavioral targeting. Lastly, we deploy our approach on live user traffic and compare its performance with conventional methods. See Section 7.

2. USER TARGETING

Advertisers are now spending increasingly larger fractions of their overall advertising budgets on online advertising. Effective online advertising is based on three key factors: the *context* in which the ad appears; the *audience* to whom the ad is targeted; and the *creative* that specifies the message being delivered via the ad. We focus on the audience for the purposes of this paper.

Next we present a generic framework for understanding audience targeting problems. Users are modeled as streams of *typed events*. The targeting system is modeled as a function defined over user histories. Within this framework the popular targeting methods such as property, user segments, and behavioral targeting are special cases.

2.1 User Modeling

We denote a user $u \in U$ as a process that emits a sequence of events $\langle e_1, \ldots, e_m \rangle$, where events $e_i \in E$ are defined as follows:

$$e \quad = \quad (t, T, p)$$

where t is the *timestamp*, T is the *type* of the event such as "search" or "page visit", and p is a *payload*. Examples of events include "issued a search query for shoes at 5PM" in which the type is "search query", the payload is "shoes" and the timestamp is 5PM.

Any targeting method can be abstracted as a function such that given a particular ad campaign a, the function outputs a binary decision as to whether or not to target the user. The function is defined as $g_a : U \rightarrow \{0, 1\}$ where for each user $u \in U$, whether the user should be targeted or not. As a targeting function, it cannot guarantee that the ad will be shown to the user. We can then abstract an ad server as a process that for a given a set of ads $a \in \mathcal{A}$, performs two main operations at each serving opportunity. First it determines the set of ads for which the user is said to qualify as follows:

$$Q(u; \mathcal{A}) \quad = \quad \{a \mid a \in \mathcal{A} \wedge g_a(u) = 1\}$$

where the function $Q : U \times \mathcal{A} \rightarrow 2^{\mathcal{A}}$ returns the set of ads which the user is allowed to see — qualify. Given this set of ads, the ad server then decides which one of these ads should actually be shown to the user, based on some set of business policies which are outside of the scope of this paper. Based on this description of an ad server, we see that the purpose of a targeting system is simply to target users prior to the actual ad serving.

2.2 Property Targeting

Property targeting is a simple, yet popular targeting mechanism. The advertiser specifies some set of pages P on which the ad should be shown. Users visiting those page are targeted with the corresponding ads. For example, an advertiser who sells cars could show ads on websites about cars.

This form of property targeting is completely independent of the user's history or state, instead it exploits exactly one feature — the page the user is visiting at this moment. The

property targeting function is defined as follows:

$$g(u) = \begin{cases} 1 & u \text{ is visiting } p \in P \\ 0 & \text{otherwise} \end{cases}$$

where p is the current webpage. The effectiveness of this model assumes that there is a causal link between the page and the user's interest.

2.3 User Segment Targeting

Marketing research and practice has a long history of advertising based on user profiles or personas [10]. Users are grouped into homogeneous segments and different segments are targeted with appropriate ads. For example, a major beverage company conducts market research for its low-calorie carbonated beverage and determine that the drink would appeal to the following segments:

- "young adults": $g_{1,1} = age(u, (15, 25))$, $g_{1,2} = country(u, \text{USA})$
- "moms": $g_{2,1} = gender(u, F)$, $g_{2,2} = age(u, (25, 45))$, $g_{2,3} = country(u, \text{USA})$

Segments are defined as logical predicates such as $age(u, (15, 25))$ which is true if the age of user u is between 15 and 25 years. Common types of segments are *demographic*, *geographic*, and *psychographic* attributes [8].

In the above example, the advertiser would define the targeting function as follows:

$$g(u) = \begin{cases} 1 & u \in S \\ 0 & u \notin S \end{cases}$$

$$\text{where } S = \bigcup_{i=1}^{n} S_i = \bigcup_{i=1}^{n} \bigcap g_{i,j}$$

where $S_i$is the i^{th}segment and $g_{i,j}$ is the set of users who qualify for the j^{th}predicate of segment S_i.

Segment targeting is popular since it is easy to understand and implement and moreover provides advertisers transparency and control over the audience selected for targeting their ads. There are two main limitations. a) The advertiser must match the ad to the pre-existing segments. b) The segments may not be expressive enough to truly capture the audience of interest to the advertiser. For example in the "moms" segment, we are missing the important attribute $hasChildren(u)$, which is clearly a defining feature of the persona.

2.4 Behavioral Targeting

Behavioral targeting provides an approach to learn the targeting function from historical data [4, 14]. In general, behavioral targeting focuses on leveraging the past behavior of a user to predict their future behavior. Although behavioral targeting methods vary in terms of the set of features, the objective optimized, they can be generalized within our targeting framework. A behavioral targeting method B can be viewed as a process that learns a targeting function $g(u)$ from data about users and ads. Formally this can be described as function $B : \{U\}^m \rightarrow \mathcal{G}$ that maps a sample of m users from the set of all users U onto a specific space of targeting functions $\mathcal{G}$ such as linear classifiers trained on feature vectors constructed from user histories.

The predictive model can leverage a very rich set of user features and can be trained to directly optimize for the performance of the ad, e.g., CTR, conversion rate etc. More importantly, the framework of the behavioral targeting model is general enough to accommodate other information such as the current context – the webpage the user is visiting, membership in various user segments, and more recently social network data [11].

Continuing with the targeting example, online behavior could be used for targeting rather than simple segments.

- "young adults": heavy usage of messenger / social networking sites, searches for common school topics.
- "moms": searching for products related to children, reading parenting articles.

The focus of this paper is on behavioral targeting that goes beyond age and gender. By training our algorithm on actual data, we can move beyond simple ad targeting based on stereotypes (showing pickup trucks during baseball games) and towards a personalized experience that can benefit the user and the advertiser alike.

3. PROBLEM DESCRIPTION

In our setting, we are trying to improve various existing campaigns where the advertisers pay per conversion, called CPA campaigns. Each campaign has already been tuned manually with user segment targeting and property targeting. However, no behavioral targeting has been done. As noted in previous work [3], our objective with this system is to refine the targeting constraints using past behavior of the user. Through refinement we can improve the number of conversions per ad impression without greatly increasing the number of impressions, which increases the value of our inventory.[1]

Note that when making a decision about whether to target an impression based on user behavior, we cannot use any information from the day of the impression or any day afterward to impact the classification of the impression. As shown in Figure 1, we consider user history as a sequence of events relative to some *target time*, τ, at which time the user is being considered for targeting. We decompose the user's sequence of events around the target time τ as follows:

$$\begin{aligned} u(\tau) &= (E_F(\tau), E_T(\tau)) \\ \text{where } E_F &= \{e \mid e \in E \wedge t(e) < \tau\} \\ \text{and } E_T &= \{e \mid e \in E \wedge \tau \leq t(e) \leq \tau + \delta\} \end{aligned}$$

Here $E_F(\tau)$ denotes the events prior to the target time which we call the *feature window* and $E_T(\tau)$ denotes the events that occur between τ and $\tau + \delta$ (where δ is 1 day) which we call the *target window*.

Hence, when we model behavioral targeting as a machine learning task where each user history forms an example: a user is a positive example if it is credited to a conversion in the target window, a negative example otherwise.[2] Given training users $\{1, \ldots, m\}$ define $T = \langle (x^1, y^1), ..(x^m, y^m) \rangle \in (\mathbb{R}^n \times \{-1, +1\})^m$ to be the training data, where x^i is a feature vector constructed from the events of the user i in

[1]Although it is interesting to imagine relaxing these constraints, that is impossible to analyze offline.

[2]If several impressions are shown to a user and the user eventually converted, no more than one impression from that user is considered a positive example.

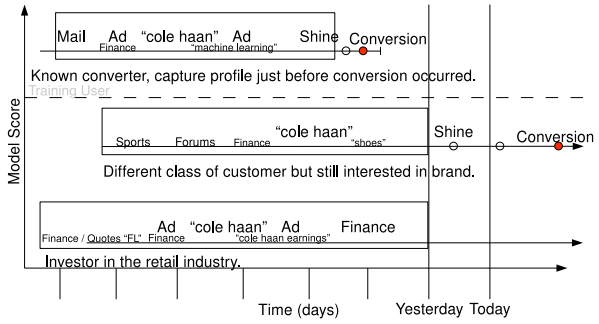

Figure 1: Targeting model is trained on user histories (rectangles) as they existed prior to the start of the conversion process (open circle) that led to the conversion (solid circle). For evaluation, all users are given the same target time (yesterday) and the ad server may choose to show ads at some point in the future to start the conversion process (open circle).

the feature window, $y^i = +1$ if the ith example is positive, and $y^i = -1$ otherwise. The test set is defined similarly.

The most significant difference between traditional targeting and this work is in exploiting the user profiles for maximizing conversions. Hence, in this paper we fix our learning algorithm to be a support vector machine [13] and focus on the issue of how to leverage user history efficiently. In particular, how to convert user profiles into feature vectors (e.g., representing different events as features, computing their weights, incorporating timestamps) and how to construct target labels (e.g., clicks, conversions). A user representation method consists of a function $\phi : U \rightarrow \mathbb{R}^n$, where $\mathbb{R}^n$ is the Euclidean space of dimension n. A target label function is defined as $\gamma : U \rightarrow \{-1, +1\}$. Given this, we extract an appropriate feature and target label set as $(x, y) = (\phi(u), \gamma(u))$. We then select a vector $w \in \mathbb{R}^n$ by solving the following optimization:

$$\arg \min_{w \in R^n} w^2 + C \sum_{i=1}^{m} L(w \cdot x^i),$$

where $L(\hat{y}, y) = \max(1 - \hat{y}y, 0)$ and C is a constant that controls the balance between regularization and minimizing the loss on the training set.

In this work, we investigate different user representation operators $\phi(u)$ and different target labeling functions $\gamma(u)$. These are summarized as follows:

- User representation function: convert events to features, compute feature weights, incorporate timestamps and history length.
- Target labeling function: clicks or conversions

4. USER REPRESENTATION

In this section we propose and empirically compare different feature extraction methods for behavioral targeting.

4.1 Data Setup

In our experiments we build targeting models for display advertising campaigns. We collected 4 weeks of data for 226 campaigns. Each campaign is treated as a separate targeting task. The first three weeks of data is used for training, while the last week is used for testing. Each example impression is preceded by at least 4 weeks of user events. In total, with some sampling of negative examples we have a total of approximately 40M examples in training and 80K of these examples are positive. This is a benchmark set that enables us to do rigorous offline experiments.

4.2 Baseline Method

In each of the experiments, we compare targeting methods in terms of the area under the ROC curve (AUC). The benchmark baseline consists of a simple application of the support vector machine classifier. Based on some simple experiments, we arrived at the following parameters. Because we have a large number of mostly irrelevant features, we choose a strong regularization parameter of $C \in [0.001, 0.05]$. Next, since we have an highly imbalanced class distribution, we choose a class weighting parameter of 10:1 cost for the positive and negative class, respectively. In order to have a single parameter configuration for all learning tasks, we sample the negative examples such that the class ratio is always satisfied. The empirical evaluation is based on this baseline. Unless otherwise specified, all metrics are measured as conversion-weighted average of AUC across all campaigns in the benchmark set.

4.3 Event Types

When collecting the user's historical online behavior, we consider both *active* and *passive* activities. Passive activities include viewing ads and visiting pages in which an action is not specifically required upon seeing the page. Active activities include issuing search queries and clicking ads in which users actually perform an action on the page. Browsing activity is somewhat active because the user took action to visit the page, but we argue that the activity is less than specifically typing a search query or clicking on an ad. We investigate which source of activity is stronger in predicting the user's propensity to convert on a set of advertisements.

The activities of the users are a sequence of events collected from server logs. Events are associated with both a timestamp and metadata. For example, an event could be a visit to a finance web page and the metadata associated with the event is the content of the page, which is logged separately and then joined with the event along with the anonymized identifier of the user and the time. We consider several different events, each with a corresponding feature extraction method.

- *Pages visited*: Website pages are clustered into several smaller subdirectories. Some features extracted are the id of the cluster and the category of the page from an existing hierarchical page categorizer.
- *Search queries*: Searches issued, clicks on search links, clicks on search advertising links. Some features extracted are the query category, the click information and the unigrams in the query (details later).
- *Graphical Ads*: Views and clicks on ads. Some features extracted are the ad category, the click information and the id of the ad (details later).

For all feature types, we use a common feature representation (i.e., weighting) operator $\phi : E^\star \rightarrow \mathbb{R}^n$ which we call

relative frequency bag of events. The frequency of an event is defined as the number of days in which the user has performed the event. We consider events separately by type (e.g., page visits, search queries). For example we concentrate on page visits and denote by event p the "visit to any e-mail page" and other page view events by q and r. If the user had visited the pages p, q, and r in a sequence over four days as follows:

$$e = (p_1, p_1, p_2, q_2, r_3, q_4)$$

where each event p_i denotes the visit on page p on day i, then the frequency bag of events representation for page visits would be: $F_p(e) = (p : 2, q : 2, r : 1)$ where we use the convention of $p : n$ to mean that the feature p has the value n in the feature vector. Here, the page p was visited 3 times but on just 2 distinct dates. The relative frequency representation normalizes within each feature type such that the final feature vector is defined as follows:

$$\phi(e) = \left(\frac{F_1(e)}{\|F_1(e)\|}, \ldots, \frac{F_n(e)}{\|F_n(e)\|} \right)$$

where n is the number of feature types such as page views, ad views, ad clicks, *etc.*

4.3.1 Browsing Activities

As users interact with a website they consume content either through direct action such as searching for information or casual browsing such as checking e-mail or reading news headlines. However, the particular stream of content consumed gives us some insight into the user' interest.

Rather than considering each individual URL, we group different URLs together and count visits on any of them. We considered 3 grouping methods: *site index*, *semantic categories*, and *server location*. The site index is based on the navigational pattern within the specific website. For example, if a user visits a page showing the profile of a company in the stock market, then the site index would be something like "stock_quotes" which is a sub-index of the main "finance" page. This reduces the set of URLs from millions into approximately 30,000. The benefit is a reduced space, but the main disadvantage is that we lose the granular information such as which stock quote the user is viewing. The semantic categories come from a manually annotated categorization system and map the pages into a taxonomy of roughly 1,000 nodes that describe the information on the page. In our example of a stock quote page, we consider the categories along the path in the taxonomy. If the category is "Finance / Stocks / Quotes", then we add the features "Finance", "Finance / Stocks", and "Finance / Stocks / Quotes" to the user's profile. With the rollup, the model should consider the hierarchical placement of the page even when the coverage is low at the leaf nodes. Finally, we reduce the space of pages even further to consider the international location of the user when he or she saw the page. For example, if the user is viewing a finance page in the US, then the feature would be "Page in US", however if user were to see the same page in Europe, then the page would be "Page in EU". This encoding of pages is at a high level of aggregation but can still tell us something about the user–language, location, etc.

Table 1 compares the improvement in average area under the ROC curve for the benchmark campaigns when considering the different browsing patterns. In the figure the user location information makes the baseline and has the least

	ΔAUC
User Location	0.0%
Only Categories	9.29%
Only Site IDs	11.92%
All Browsing Activity	15.53%

Table 1: Relative difference in performance relative to user location when adding more browsing activity.

	ΔAUC
Raw Ads	0.00%
Categorized Ads	0.01%
Raw + Categorized Ads	0.39%

Table 2: Relative difference with respect to baseline when adding more ad activities.

information , but we note that it is significantly better than random targeting despite being very coarse. The site index grouping performs very well as a single source of page visits, slightly better than the semantic categories. We think that the semantic categories can be too coarse as a 30:1 decrease in the number of features. Overall, however, adding all page browsing features is best, which indicates that the different aggregations bring some useful signal.

4.3.2 Ad Activities

When users visit pages, they typically interact with graphical ads either by clicking or simply viewing ads. Although ad clicks are clearly useful features in predicting ad activities, the ad views may also be useful. To see this, we consider the following example, an advertiser targets users who previously visited their home page and the ad is shown to these users anywhere they go (this is a common targeting method). If the only way that the users have seen this ad is by going to the advertiser's site in the past, we can infer that if the user saw the ad, he or she must have visited the advertiser's site regardless of whether we have the browsing history.

Like the browsing activity features, we consider multiple groups of ad features. Ads are placed on pages in prespecified positions such as on the left or right sides and in different sizes such as 300×250-pixel rectangles or tall vertical bars. Of course each ad is a different image and may be a static image, video, or flash. All of these features influence the click rate on the ad as well as the degree to which the user notices the ad. In order to capture all the factors that influence ad interest, we denote an ad by a triplet of: creative, position id, and targeting parameters. Each of these elements is denoted by a unique identifier. Like the browsing history, we consider both the clicks and views for ad triplets as well as clicks and views on semantic categories of ads.

Table 2 shows the relative improvement of adding categorized ads to raw ads. We see that there is not much difference in performance for ad triplets and categorized ad activities when treated separately. These results indicate that categorized ads do provide some signal in addition to just the raw ad activities.

4.3.3 Search History

Presumably the most direct user behavior is a search query because users have to manually type in the query. The po-

	ΔAUC
Raw Queries	0.00%
Categorized Queries	-0.29%
Raw + Categorized	0.13%

Table 3: Relative difference with respect to base when adding more query features.

	ΔAUC
User Location	0.00%
Only Browsing Activity	15.53%
Only Ads Activity	18.23%
Only Query Activity (*)	18.34%
Browsing + Ads + Query	18.23%

Table 4: Relative difference in feature types. For query activity (*), we note that the testing set is a subset of the larger set.

tential disadvantage to queries is that they are typically very short and are very ambiguous, containing misspellings, synonyms, different spacing etc. We consider three types of behavior for queries: simply issuing a query, clicking on a link, and clicking on a search ad on the results page for the query. We group all these activities together and evaluate whether they are predictive of conversions. We group queries together into semantic categories using a machine learned query categorizer. We consider both categorized and raw queries. For the raw queries, we split on white space to create unigrams.

Table 3 shows the relative improvement from adding categorized query features to raw queries. There is a performance decrease from considering only the categorized queries versus raw queries. This is interesting when we consider that there are only about 1,000 categories but millions of unique queries issued by the users. This suggests that the categorization of the queries may be noisy or incomplete. However, it seems that when available the categorized queries help in conjunction with the raw queries.

4.3.4 Discussion

We see that within each feature type, categorized features alone do not perform as well as the raw features. However, the best strategy seems to be to combine raw and categorized features. We now consider adding all the feature types together. Table 4 shows that, relative to user location, considering each feature type alone does very well. In addition, adding all feature types does very well. Of all the features we considered, ads and browsing activity appear to be the most informative.

In Table 4 we see that queries show the largest improvement in performance; however, we argue that this is misleading. To better understand the results, we consider the coverage of the different feature types. We define coverage as the proportion of users that have the feature defined over the set of all users. From our dataset we see that all users have some browsing activity, and nearly all users have some ad activity. However, a smaller proportion of users have search activity. And the fewest users have some ad click activity. This indicates that the most intuitively informative features: searches and ad clicks have the lowest coverage. Hence, although they are very informative and might have good performance when available, their overall impact is lower. Based on the coverage analysis, we see that even if browsing activity is less informative when users have ad click or query activities, when we consider the union of all activities, those which occur most frequently are preferred– especially in regularized learning algorithms.

Table 5 shows the relevant features from 3 example campaign models. These features support our hypothesis that the baseline approach of adding all features is better overall. In both the university model and the airline, we see that searches are relevant. In addition, these searches are very relevant to the subject of the ad such as air travel or airports. However in the home telecom model, we see that ads are the most relevant features. To see why ads are most relevant for home telecom versus the other campaigns is that for the telecom service there are no good search terms that indicate interest. Instead, the main customers are people who own homes or rent apartments. Viewing health-related ads and checking news and mail are activities common to homeowners — the events that describe the persona of the target audience.

4.4 Feature Representations

In Section 4.3, we use L_2 normalized frequency across each feature group (e.g., page visits and search queries) to define a baseline feature representation. In the following section, we discuss an alternative representation based on statistical measures of how predictive individual features are of conversion events.

4.4.1 Log-Likelihood Ratios (LLRs) as Features

We tried using a feature representation based on log-likelihood ratios (LLRs) of raw features occurring in conversion events. This approach is based on the campaign-dependent feature representation described in [7]. In this representation, ad campaign i extracts feature vectors of the form, $\psi_i\left(e^j\right) \equiv \left(\psi^j_{i,0},\ldots,\psi^j_{i,N}\right)^T$, where e^j is an input example. These features are derived from the corresponding baseline feature vector of Section 4.3: $\phi\left(e^j\right) \equiv \left(\phi^j_1,\ldots,\phi^j_N\right)^T$. If baseline feature ϕ^j_n exists in example e^j— that is, if it has a non-zero count — then feature $\psi^j_{i,n}$ is set equal to the log-likelihood ratio of ϕ_n occurring in a conversion event for ad campaign i. We use $c\left(a_i\right)$ to represent a conversion event on ad campaign i (conversely, $\overline{c}\left(a_i\right)$ represents a non-conversion event). Feature $\psi^j_{i,n}$ can now be expressed mathematically as follows:

$$\psi^j_{i,n} \equiv \mathbf{1}(\phi^j_n > 0)\cdot\log\frac{Pr\left(\phi_n > 0|c\left(a_i\right)\right)}{Pr\left(\phi_n > 0|\overline{c}\left(a_i\right)\right)} \quad \forall n \in \{1,\ldots,N\}. \tag{1}$$

We also define feature $\psi^j_{i,0}$ as

$$\psi^j_{i,0} \equiv \log\frac{Pr\left(c\left(a_i\right)\right)}{Pr\left(\overline{c}\left(a_i\right)\right)}. \tag{2}$$

It can be shown that if the baseline features are conditionally independent of one-another given $c\left(a_i\right)$, then the following is true [7]:

$$\sum_{n=0}^{N}\psi^j_{i,n} = \text{logit}\big(Pr\left(c\left(a_i\right)|\Phi_+\left(e^j\right)\right)\big),$$

where $\Phi_+\left(e^j\right)$ is defined as

$$\Phi_+\left(e^j\right) \equiv \left\{\mathbf{1}\left(\phi^j_1 > 0\right),\ldots,\mathbf{1}\left(\phi^j_N > 0\right)\right\}.$$

#	University Model	#	Low-Cost Airline	#	Home Telecom
5	Social Networking Search	1	Mail Page	1	Groups Page
2	Mail Pages	3	Generic Search Activity	4	News Ads/Page
7	Credit-Score Ads	4	Brand-relevant terms	6	Credit Ads
2	Search "university"	3	Credit-score ads	4	Car/Health Ad
4	financial aid searches	5	Search Results on Air Travel	1	Mail Page
		2	Search for Airports	3	Technology Ad

Table 5: Number of features in each type for three selected campaigns.

	ΔAUC
large campaigns	+2.25%
medium campaigns	-0.89%
small campaigns	-2.11%

Table 6: Gain in AUC for LLR-based feature representation relative to baseline.

Since summing over $\left\{\psi^j_{i,0},\ldots,\psi^j_{i,N}\right\}$ yields the logit of $Pr\left(c\left(a_i\right)|\Phi_+\left(e^j\right)\right)$, it follows that if the $\Phi_+\left(e^j\right)$ features are conditionally independent of one-another given $c\left(a_i\right)$, then $\psi_i\left(e^j\right)$ is the globally optimal representation of $\Phi_+\left(e^j\right)$ (note, however, that this optimality depends on one's ability to exactly compute the LLRs in (1) and (2); this is often not possible in practice). One can further show that the following linear decision function is equivalent to a Naive Bayes classifier trained on $\Phi_+\left(e^j\right)$ if the weight vector w in the following expression is composed of all ones (i.e., $w = (1,\ldots,1)^T$) :

$$\hat{y}_i\left(e^j\right) \equiv \begin{cases} 1 & w^T\psi_i\left(e^j\right) > 0 \\ -1 & otherwise \end{cases}$$

Here, $\hat{y}_i\left(e^j\right)$ represents the hypothesized class (1 or -1) of example e^j in ad campaign i. In our experiments, we estimated the probabilities in (1) and (2) empirically from the training data. We used the following count threshold when estimating probabilities: For all ϕ_n that appear in fewer than 5 conversion examples in ad campaign i, we set $\psi^j_{i,n} \leftarrow 0$ for all j.

Table 6 shows the AUC results for the LLR-based feature representation relative to the baseline. We divided our campaigns into three categories, large, medium and small. The large group contains the top one-third of the campaigns with the most number of conversions, while the small group contains the bottom one-third. These results show that the LLRs tend to outperform the baseline representation on large campaigns but not on small campaigns. One explanation for the reduced performance on small campaigns is that the probability estimates in Equation 1 tend to be noisy when only a small number of events are available. We believe that it may be possible to improve the performance of the LLR-based representation on all campaigns by using smoothed probability estimates when computing the LLRs. We leave this as future work.

5. RELATION BETWEEN CLICKS AND CONVERSIONS

As mentioned before, behavioral targeting can be optimized using a variety of metrics, e.g., number of ad impressions (CPI), clicks obtained (CPC), number of conversions made (CPA), revenue earned. While CPA (revenue) is the end goal, clicks (CPC) are often used for evaluation since they are easy to instrument and measure. In this section we study the relation between clicks and conversions.

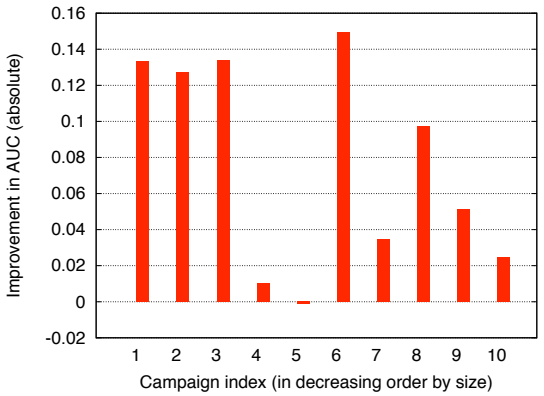

Figure 2: Improvement in prediction accuracy by using conversions for training instead of clicks. Testing is done using conversions in both cases.

5.1 Using Clicks in Place of Conversions

We start by performing the following experiment: use clicks to predict conversions and see how it compares to using conversions. In other words, in one case we train the model using clicks in the training set while in the other case the model is trained using conversion examples. Of course, the testset is labeled by conversions in both cases. As before we use support vector machines for training the models.

Clearly, using conversions for both training and testing is naturally superior. But if this hypothesis is true that clicks are "well aligned" with conversions, then using clicks for training should also perform comparable.

In Figure 2 we show the improvement in prediction accuracy achieved by using conversions for training, in comparison to using clicks for training, on 10 large campaigns. The y-axis is the difference between the performance of using conversions and using clicks (in terms of the area under the ROC curve for a campaign). Except on campaign index 4 and 5, on other campaigns using clicks performs significantly worse than using conversions. This implies that targeting users based on clicks does not necessarily mean maximizing for conversions.

5.2 Using clicks in Conjunction with Conversions

From the previous experiment we found that clicks and conversions are related but cannot be substituted for each other. In other words, for predicting conversions it is better

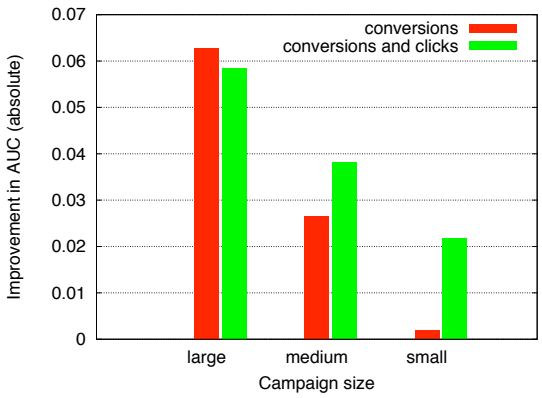

Figure 3: Improvement in prediction using clicks in conjunction with conversions.

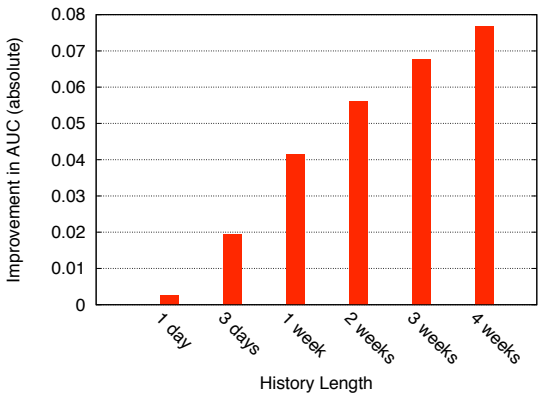

Figure 4: Effect of the length of user history.

to train the model using conversions. However, in a practical setting we often lack positive examples to train conversion models for two reasons: (a) clicks are known to be rare and conversions are even rarer, (b) due to business constraints, examples from different campaigns are not allowed to be mixed together. As a result, while we can afford to learn conversion models on large campaigns, for the medium and small campaigns we face great difficulty in doing so. Also, when the model is learned using a small number of positive examples, it often does not perform well on the testset.

To deal with this challenge, we tried a hybrid approach whereby we use clicks, which are much more abundant compare to conversions, in conjunction with conversions to train models. In particular, we treat those examples as positives which contained any of the two, a conversion(s) or a click(s). It makes sense because clicks hint of positive intent of the user. Hence, instead of grouping them with negative examples, perhaps we can generalize from them to identify other potentially interested users.

Next we compare the performance of the following two approaches: (a) label positive examples for training using conversions only and (b) using both clicks and conversions. The results are shown in Figure 3. We divided our campaigns as in Section 4.4 into three categories, large, medium and small. We treat the performance of the conversions-only approach on the small group as the baseline and show other performance numbers with respect to this baseline.

We notice that the performance of conversions-only approach worsens as we go from the large to the small group. This is expected because as the positive examples get rare, it becomes difficult to train models and perform well. We note that in the medium and small campaigns, by using clicks we can improve the performance significantly. This validates how clicks examples show positive intent of the user and when conversions are scarce, these examples can help in generalizing the model.

When there is abundant positive conversion data, like the campaigns in the large group, click examples do not help. In fact, they make the model slightly worse, which is in line with our observation from the previous section (Figure 2).

6. UNDERSTANDING THE TEMPORAL ASPECT OF USER HISTORY

In this section we focus on understanding the temporal aspect of user history. In particular, we investigate how far we should track back the user history before it stops paying off in terms of prediction performance. Also, we explore whether recent activities are more indicative of online purchases than the older ones.

6.1 Effect of History Length

In this experiment we study the effect of history length on prediction performance. More specifically, when history length is set to l days, we take into account the user history from $[\tau - l, \tau]$ period to make the prediction where τ is the target time. We treat the one day history length ($l = 1$) as the baseline and show other performance numbers with respect to this setting. The results are shown in Figure 4. We note that using the recent two weeks of user history performs significantly well. However, to our surprise, we note that we get a substantial improvement by extending the history from 2 weeks to 4 weeks.

These results show that a short recent history may not be enough for behavioral targeting. A reason behind this is that, broadly speaking, users make two kinds of conversions/online purchases on the web: (a) transactions which are quick and do not take much thought such as purchasing a movie ticket (**short-term conversions**) and (b) transactions which span a longer period of time such as car purchase or online education enrollment, and can often take several days/weeks for the user to make up her mind before making the purchase/conversion (**long-term conversions**). While short-term conversions can be made using a short history (of 1 or 2 weeks), in order to be able to predict long-term conversions, a longer history of user's activities is essential.

Longer history can also help with short-term conversions because more data allows better inferencing and refinement of user interests, e.g., within movie enthusiasts distinguish between those interested in horror versus romantic movies. As a result, longer history leads to better targeting of users, in general. However, beyond a certain history length the improvement obtained (in terms of revenue dollars) by extending history might surpass the cost of maintaining the systems storing the data. Hence, in a practical setting one might have to truncate the history beyond a certain length.

6.2 Effect of Recency

From the previous section we concluded that longer user history leads to better performance. In this experiment we

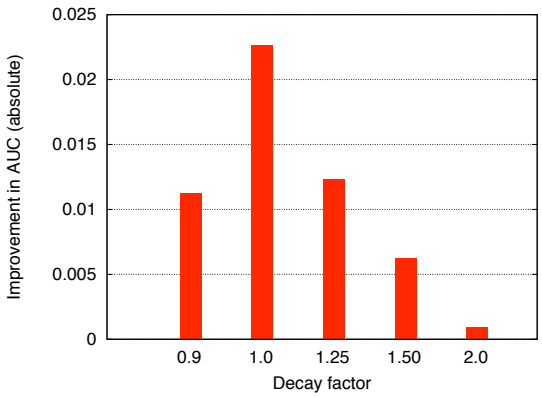

Figure 5: Effect of recency.

investigate whether by giving more weight to the recent activities in comparison to the older ones, we can further improve the performance.

As described before (Section 4.3), we set the weight of each feature in user profile based on the number of days on which the feature is present. Hence, we do not not distinguish between the features which occurred in recent days and those which had occurred earlier. For this experiment, to give more weights to recent features we set the weight of feature f to:

$$\sum_{t_i \in <f,u>} \alpha^{-(\tau - t_i)}$$

where τ is the target time (when the prediction is being made), α is the decay factor and (f, u) is the sequence of days on which feature f is present in the history of user u. When $\alpha = 1$ the above weighting approach reduces to uniform weighting. By setting α to a larger constant, we can give more weight to recent user activities in comparison to the older ones. (The weights are then normalized within each feature type, as before.)

In Figure 5 we show the effect of decay constant α on the prediction performance. (We treat $\alpha = 2$ as the baseline and report other performance numbers with respect to it.) For the sake of completeness, we also show $\alpha < 1$ which favors older features compare to the recent ones. It is clear that the best performance is achieved by uniform weighting (i.e., $\alpha = 1$). This is somewhat counter-intuitive since we expect recent activities to be a better indicator of user interests.

Given our experiments from Section 6.1, a possible explanation for this result can be that decay effectively shortens the user history. For example, when $\alpha = 2$ a feature that is 10 days old gets a weight of smaller than 10^{-3}, hence the history length is practically reduced to 10 days (or smaller). As a result, long-term conversions (explained earlier) become more difficult to predict. In other words, there is a trade-off between recency and history length. While the former might help in predicting short-term conversions, the latter is needed for long-term conversions.

To further validate this finding, we tried another approach for taking recency into account. For this experiment we divided the user history into multiple buckets based on time periods, e.g., activities from the last day are put into one bucket, activities from 2-7 days in the next bucket and so on. We labeled the features with the buckets they occurred in and let the SVM learn weights for different (feature, bucket) pairs. Unlike the previous experiment where we were confined to exponential decay, this method allows us to automatically learn non-uniform weighting of features to account for recency.

However, similar to the decay experiment, we found that the results from this bucket experiment were worse than the uniform weighting approach.

	Δ Conversion Rate	Δ eCPA
Campaign 1	+80%	-37.55%
Campaign 2	+57.14%	-27.39%
Campaign 3	+264%	N/A
Campaign 4	+83%	N/A

Table 7: Gain in conversion rates and eCPA values over existing CTR-based targeting methods in live experiments.

7. COMPARISON WITH EXISTING TARGETING IN LIVE EXPERIMENTS

Driven by the good performance of our behavioral targeting approach in offline experiments, we tested the system on live traffic on a few ad campaigns on a major US advertising network. We trained our models using the practices described in the previous sections. We compared our models to those trained by an existing behavioral targeting system (as in [4]): (a) which targets users whose activities/interest in the advertiser's category is above a certain threshold, (b) the model optimizes for click-through rates.

For training/scoring the models we generated user profiles spanning 8 weeks of user history. We scored the models on a daily basis and the experiment was run live for three months. For each of the campaigns, each of the existing and new models received at least a total number of 1M impressions on a monthly basis.

Table 7 shows the overall performance of our models as compared to the original CTR-optimized behavioral targeting models. In terms of the conversion rate, i.e., the percentage of viewers that convert, we note that our models achieved significant improvements compared to the existing models (while keeping the same coverage). This demonstrates the ability of our models to target users with much higher tendency to convert than existing CTR-optimized models. Another important metric that we report is the Effective Cost Per Action (eCPA), used to measure the effectiveness of the inventory purchased by the advertiser. Effectively, the eCPA tells the advertiser what they would have to pay for each conversion. Our results show a considerable decrease in eCPA that our models achieved compared to the other models (we could not report eCPA values for campaign 3 and 4 for business reasons). This is considered as a major gain from the point of view of the advertisers, as this means that we are able to reach the audience desired for the campaign while decreasing the amount of impressions (i.e., retries) needed. Both these results demonstrate the effectiveness of our models in large-scale behavioral targeting.

8. RELATED WORK

In this work, we use behavioral targeting to improve the conversion rate. Targeting users who will convert is a difficult problem: often, the problem is divided into predicting clicks, and predicting the probability that the click will con-

vert [6, 1, 12]. The advantage of this division relates to business logic: the publisher (such as Yahoo! or Google) has data about how likely users are to follow various paths towards clicks on advertisements on their site. On the other hand, advertisers have more information about the paths of users on their website. Therefore, there is a certain cleanliness with regards to data ownership.

On the other hand, paying for conversions has two effects. First of all, there is the maximum level of quality control of the traffic. Problems such as click fraud [5, 9, 15] do not arise. Second, when creating a conversion model, one is aware of the abundance of users who did not click. This plethora of negative data can really help: intuitively, knowing that someone was unlikely to click makes it quite possible that they are unlikely to convert.

In this paper we focused on building such conversion models. We compared our approach with existing behavioral targeting methods (such as [4, 14]) which optimize for click-through rates and showed how optimizing directly for conversions can lead to improved performance. Compared to previous work on conversion optimization [1, 2, 3, 6, 12], our work makes several new contributions: we look into understanding the effect of different user activities on prediction, give insights about the temporal aspect of user behavior (recency vs. long-term trends) and explore different variants (user representation and target label) through large offline and online experiments.

9. CONCLUSION

Determining which users are most likely to respond to a given advertisement is the primary goal of targeting. By using the user's historical online behavior, behavioral targeting can greatly improve in terms of performance and reach relative to hand-tuned segments. In training models for behavioral targeting, we are concerned with both what to predict and how to leverage user profiles. Our empirical analysis of over 200 advertising campaigns has yielded some useful best practices for targeting.

First, we compare whether training on clicks or conversions is best. Our results indicate that clicks often suffice but they are not always good proxies for conversions. By predicting directly for conversions, we can now improve the impact of targeting methods on ad campaigns. Most existing behavioral targeting systems are limited to clicks because it is the only data readily available. But, recently, advertisers have been willing to share individual responses to ads [3], facilitating such conversion-optimized models.

Second, of critical importance to behavioral targeting is the online behavior of the users. Although the behavior we can observe is only passive, we find there is considerable signal in the profile. We first consider the types of events: browsing, ads, or query activity. The results indicate that although query activity is quite useful, browsing activity is more valuable overall. Then given a set of selected user events, we consider a feature representation (LLRs) that encodes the correlation between feature and conversion. We found that using LLRs improves performance on large campaigns but not on small campaigns. Next we turn to the amount of user history to use for prediction. The results indicate that more data is better, but there is a point of diminishing returns of roughly 30 days which is primarily due to the changing nature of user interests and web sites. We next find that rather than explicitly incorporating temporality into the features, simply aggregating events performs well.

Lastly, based on extensive offline and online experiments, we validate our findings and have arrived at several best practices for behavioral targeting.

10. REFERENCES

[1] N. Archak, V. S. Mirrokni, and S. Muthukrishnan. Mining advertiser-specific user behavior using adfactors. In *Proceedings of the 19th International World Wide Web Conference*, 2010.

[2] A. Bagherjeiran, A. O. Hatch, and A. Ratnaparkhi. Ranking for the conversion funnel. In *Proceeding of the 33rd SIGIR conference on Research and development in information retrieval*, 2010.

[3] A. Bagherjeiran, A. O. Hatch, A. Ratnaparkhi, and R. Parekh. Large-scale customized models for advertisers. In *ICDM Workshops*, 2010.

[4] Y. Chen, D. Pavlov, and J. Canny. Large-scale behavioral targeting. In *Proceedings of the 15th SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009.

[5] I. Click Forensics. Click fraud index. `http://www.clickforensics.com/resources/click-fraud-index.html`, 2010.

[6] Google, Inc. Google analytics. `http://www.google.com/analytics`.

[7] A. Hatch, A. Bagherjeiran, and A. Ratnaparkhi. Clickable terms for contextual advertising. In *ADKDD*, 2010.

[8] I. Nielsen Company. Nielsen Claritas PRIZM. `http://en-us.nielsen.com/tab/product_families/nielsen_claritas/prizm`.

[9] Y. Peng, L. Zhang, M. Chang, and Y. Guan. An effective method for combating malicious scripts clickbots. In *Proceedings of the 14th European Symposium on Research in Computer Security*, 2009.

[10] B. J. Pine. Mass customizing products and services. *Strategy & Leadership*, 21(4):6 – 55, 1993.

[11] F. Provost, B. Dalessandro, R. Hook, X. Zhang, and A. Murray. Audience selection for on-line brand advertising: privacy-friendly social network targeting. In *Proceedings of the 15th SIGKDD international conference on Knowledge discovery and data mining*, 2009.

[12] B. Rey and A. Kannan. Conversion rate based bid adjustment for sponsored search auctions. In *Proceedings of the 19th International World Wide Web Conference*, 2010.

[13] X.-R. Wang, K.-W. Chang, C.-J. Hsieh, R.-E. Fan, G.-X. Yuan, H.-F. Yu, F.-L. Huang, and C.-J. Lin. Liblinear – a library for large linear classification. `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`.

[14] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen. How much can behavioral targeting help online advertising? In *Proceedings of the 18th International Conference on World Wide Web*, 2009.

[15] L. Zhang and Y. Guan. Detecting click fraud in pay-per-click streams of online advertising networks. In *Proceedings of the 28th IEEE International Conference on Distributed Computing Systems*, 2008.